

**WHAT IS CLAIMED IS:**

1. A method for optimizing a run time for an object code generated from a source code, the method comprising:
  - extracting information for each procedure call contained in the source code;
  - selecting a call linkage between a caller procedure and a callee procedure for each procedure call using the extracted information, where the selected call linkage is optimized to minimize a run time of an object code generated from the source code;
  - generating the object code from the source code; and
  - running the object code using the selected call linkages for each procedure call.
2. The method of claim 1 wherein the determined call linkage is one of a memory-based call linkage and a register-based call linkage.
3. The method of claim 2 wherein, if the memory-based call linkage is selected for a particular procedure call, the running comprises:
  - allocating a block in a memory to store a value for each argument in the particular procedure call;
  - storing the value for each argument from a register in a processor to the block in the memory;
  - branching the procedure call to a callee procedure; and
  - loading the value for each argument from the block in the memory back to the register.
4. The method of claim 2 wherein, if the register-based call linkage is selected for a particular procedure call, the running comprises:
  - copying a value, for each argument in a procedure call, from a register in the processor to a parameter register in the processor;
  - branching the procedure call to a callee procedure; and
  - copying the value from the parameter register back to the register.
5. The method of claim 2 wherein the selecting comprises:
  - detecting whether an error exists for the procedure call;
  - selecting the memory based call linkage if the error is detected for the

procedure call; and

selecting the registered based call linkage if no error is detected for the procedure call.

6. The method of claim 5 wherein the error is detected if the procedure call has a different number of parameters than the callee procedure.

7. The method of claim 5 wherein the error is detected if a parameter type for a parameter in the caller procedure is different than the parameter type for the parameter at a corresponding position in the callee procedure.

8. The method of claim 5 wherein the error is detected if a number of arguments in the procedure call is greater than a number of parameter registers used to run the object code.

9. The method of claim 5 wherein the error is detected if an argument in the procedure call is unpassable in a register.

10. The method of claim 1 wherein the extracting of procedure call information comprises:

extracting information for each procedure definition contained in the source code.

11. The method of claim 10 wherein the extracted procedure call information comprises an identifier for a calling procedure and a callee procedure, and the extracted procedure definition information comprises a number of arguments received by the procedure and a classification for each argument.

12. The method of claim 1 wherein the call linkage is selected in a class comprising one of a register stacks call linkage, a system call linkage and a near versus far call linkage.

13. The method of claim 1 wherein the extracted information is generated in a data structure used to select the call linkage for each procedure call.

14. An apparatus for optimizing a run time of an object code generated from a source code, the apparatus comprising:

a memory for storing a compiler program; and

a processor comprising a plurality of registers, where a subset of the plurality of registers comprise parameter registers, the processor performing a method upon executing the compiler program in the memory, the method comprising:

extracting information for each procedure call contained in the source code;

selecting a call linkage between a caller procedure and a callee procedure for each procedure call using the extracted information, where the selected call linkage is optimized to minimize a run time of an object code generated from the source code; and

generating the object code from the source code.

15. The apparatus of claim 14 wherein the object code is executed using the selected call linkage for each procedure call.

16. The apparatus of claim 14 wherein the determined call linkage is one of a memory-based call linkage and a register-based call linkage.

17. The apparatus of claim 16 wherein the selecting comprises:

detecting whether an error exists for the procedure call;

selecting the memory based call linkage if the error is detected for the procedure call; and

selecting the registered based call linkage if no error is detected for the procedure call.

18. The apparatus of claim 14 wherein the extracting of procedure call information comprises:

extracting information for each procedure definition contained in the source code.

19. The apparatus of claim 18 wherein the extracted procedure call information comprises an identifier for a calling procedure and a callee procedure, and the

extracted procedure definition information comprises a number of arguments received by the procedure and a classification for each argument.

20. The apparatus of claim 14 wherein the extracted information is generated in a data structure used to select the call linkage for each procedure call.

21. A computer readable medium storing a software program that, when executed by a computer, causes the computer to perform a method comprising:

- extracting information for each procedure call contained in a source code; and
- selecting a call linkage between a caller procedure and a callee procedure for each procedure call using the extracted information, where the selected call linkage is optimized to minimize a run time of an object code generated from the source code;
- generating a object code from the source code; and
- running the object code using the selected call linkages for each procedure call.

22. The computer readable medium of claim 21 wherein the determined call linkage is one of a memory-based call linkage and a register-based call linkage.

23. The computer readable medium of claim 22 wherein, if the memory-based call linkage is selected for a particular procedure call, the running comprises:

- allocating a block in a memory to store a value for each argument in the particular procedure call;
- storing the value for each argument from a register in a processor to the block in the memory;
- branching the procedure call to a callee procedure; and
- loading the value for each argument from the block in the memory back to the register.

24. The computer readable medium of claim 22 wherein, if the register-based call linkage is selected for a particular procedure call, the running comprises:

- copying a value, for each argument in a procedure call, from a register in the processor to a parameter register in the processor;
- branching the procedure call to a callee procedure; and
- copying the value from the parameter register back to the register.

25. The computer readable medium of claim 22 wherein the selecting comprises:  
detecting whether an error exists for the procedure call;  
selecting the memory based call linkage if the error is detected for the  
procedure call; and  
selecting the registered based call linkage if no error is detected for the  
procedure call.
26. The computer readable medium of claim 21 wherein the extracting of procedure  
call information comprises:  
extracting information for each procedure definition contained in the source  
code.
27. The computer readable medium of claim 26 wherein the extracted procedure  
call information comprises an identifier for a calling procedure and a callee procedure,  
and the extracted procedure definition information comprises a number of arguments  
received by the procedure and a classification for each argument.
28. The computer readable medium of claim 21 wherein the extracted information is  
generated in a data structure used to select the call linkage for each procedure call.